We thank all the reviewers for helpful comments and insights. We will incorporate suggestions about writing, diagrams and presentation in the final version of the paper. In what follows we include experiments requested by the reviewers, albeit on a small scale (80 matrices for training and 20 for testing), so the numbers might be slightly different from those in the paper. We will include full scale experiments in the final version of the paper.

To Reviewer 1:

- **Error bars and standard errors.** Currently the error is averaged over multiple test matrices (as described in lines 208-218), but the matrix is learned/generated only once. For this rebuttal we computed the matrix 5 times, on Logo, with $k = 10, m = 20$ and 100 matrices. The variance of our learned algorithm is indeed lower.

	Learned	Random
Avg Test Error	0.0629	2.264
Standard Error	0.0013	0.163

- **Complexity of the prior/proposed algorithms.** Our main contribution is a learning-augmented version of the streaming algorithm due to Sarlos/Clarkson-Woodruff (SCW). Since we augment a streaming algorithm, our main focus is on improving its *space* usage (which in the distributed setting translates into the amount of communication). The latter is $O(md)$, the size of SA. Since we optimize the tradeoff between m and the accuracy, we directly improve the space usage and/or accuracy of the algorithm.

  Regarding the *time* complexity of SCW algorithm, it is $O(nmd)$ assuming $k \le m \le \min(n, d)$.
- **Line 156-158.** SCW proposed Alg 1 with random sketching matrices S. In this paper, we follow the same framework but instead we use learned (or partially learned) matrices. Our empirical results show that the **performance**–the quality of the returned rank-k matrix–of our algorithm (i.e., Alg 1 with learned matrices) is better than of the SCW algorithm when the training distribution is close to the test distribution.

To Reviewer 3:

- **Use three video sets to learn** S. Yes, right now we learn different S for each data set. This reflects our intended applications such as processing satellite imaging data or video monitoring frames, where the data distribution does not change much between uses.

  We have run a small experiment to learn a *single* matrix S with $m = k = 10$ (300 matrices) on Logo, Eagle and Friends (100 matrices each). We then tested it on Logo and compared to a learned S using only using 100 Logo matrices, as well as to a random S. The mixed case is close to Logo and much better than Random.

	Logo+Eagle+Friends	Logo only	Random
Test Error	0.67	0.27	5.19

- **Vary the ratio of random vs learned rows.** Alas, there is no free lunch. If the number of learned rows decreases, the test error will increase. We have run a small experiment, with $m = 20, k = 10$ using 100 Logo matrices, running for 3000 iterations. See the results below. The error curve seems monotone, and there exists a tradeoff between #learned rows and the error.

#Learned Rows	1	2	3	5	8	12	15	17	18	19
Test Error	1.43	1.18	1.06	0.783	0.281	0.119	0.090	0.081	0.064	0.068

- **Worst case construction.** As suggested, we tried to use S computed for Logo, and test it on Eagle. In this experiment, we use $m = k = 10$ and 100 matrices. The sketch matrix learned using Logo works worse on Eagle, as expected, but still works better than Random. This could be because both Logo and Eagle are video datasets.

	Logo-learned	Eagle-learned	Random
Test Error	0.68	0.16	7.38

- **Common structure on** S**.** This is a good question! We have investigated it, but haven't got clear answers so far.

To Reviewer 5:

- **Running time.** We report them for Logo matrices with $m = k = 10$. It can be seen that sketching methods are much faster than the "standard" SVD. Notice that training only needs to be done *once*, and can be done offline.

SVD	SCW	Our-Inference	Our-Training
2.2s	0.03s	0.03s	9481.25s

- **Experiment over RPCA.** Thank you for your suggestion! Unfortunately, we did not get enough time to run the RPCA experiment.